UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/790,605	03/01/2004	Brian D. Harry	MS307018.1/MSFTP572US	9605

27195 7590 02/07/2008

AMIN. TUROCY & CALVIN, LLP
24TH FLOOR, NATIONAL CITY CENTER
1900 EAST NINTH STREET
CLEVELAND, OH 44114

EXAMINER	
STEELMAN, MARY J	

ART UNIT	PAPER NUMBER
2191	

NOTIFICATION DATE	DELIVERY MODE
02/07/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com
hholmes@thepatentattorneys.com
osteuball@thepatentattorneys.com

Office Action Summary	Application No.	Applicant(s)
	10/790,605	HARRY ET AL.
	Examiner	Art Unit
	MARY STEELMAN	2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 November 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/ are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to Remarks and Amendments received 11/15/2007. Per Applicant's request, claims 1, 7-14, 16-17, 22 and 23 are amended. Claims 1-23 are pending.

Claim Objections

2. In view of Applicant's amendments and comments regarding claim 1, the prior objection is hereby withdrawn.

Claim 23 recites "The user interface of claim 21...", should be –The user interface of claim 22...--

Claim Rejections - 35 USC § 112

3. In view of the amendments to claims 9-12, the prior 35 USC 112 2nd paragraph rejections are hereby withdrawn.

Claim Rejections - 35 USC § 101

4. In view of the amendments to claims 1-12, 14-15, and 22-23, the prior 35 USC 101 rejections are hereby withdrawn.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over USPN 5,752,245 to Parrish, in view of "Palantir: Raising Awareness among Configuration Management Workspaces", by Anita Sarma, Zahra Noroozi, and Andre van der hock (2003 IEEE) (hereinafter Sarma).

Per claim 1:

A software development system embodied on a computer-readable storage medium, comprising:

-a shelving component that captures a current state of an intermediate software design containing pending changes developed on a private workspace, the shelving component removes the pending changes from the software design on the private workspace after the current state has been captured;

-a version control component that processes the intermediate software design as a completed software design.

Parrish discloses:

Column 7:66 - Column 8: 2, "A Project History server manages a single history database which is responsible for maintaining current drafts and histories of program components which are part of the client Project. " Column 9: 46-48, "In this case, the Project assumes that the desired version of the component is the one in the current configuration. "

Parish failed to explicitly disclose:

pending changes developed on a private workspace, the shelving component removes the pending changes from the software design on the private workspace after the current state has been captured;

However, Sarma disclosed a "configuration management workspace awareness tool" (p. 1, right col., 3rd paragraph). Sarma disclosed workspace events (page 4, right col.) that describe ongoing activities in each workspace. Such states disclosed are POPULATED, CHANGESIN PROGRESS, CHANGESCOMMITTED, and UNPOPULATED. "The pair ChangesInProgress and ChangesCommitted may be repeated if a developer continues to make changes to the artifact before removing it from the workspace. (remove pending changes from software design on the private workspace after the current state has been captured).

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Parrish, using the teachings of Sarma, because Parrish (col. 3: 25-32) recognized the need for a program development management system which maintains configuration and revision information and which can store different code versions developed over time, using a distributed network. Likewise, Sarma (Abstract) disclosed a novel workspace awareness tool, providing developers with insight into other workspaces. One would be motivated to reveal relevant information in a distributed configuration management to improve understanding among developers.

Per claim 2:

-one or more code development systems that are employed by a developer to design software applications.

Parrish: one or more code development systems that are employed by a developer to design software applications. (See Column 3: 41-43, "A program developer, upon logging into a client terminal on the network, establishes a workspace or project and connects with one of the servers. "; see also Column 1:17-20, "This invention relates generally to improvements in computer systems and, more particularly, to object-oriented software for managing changes, revisions, and modifications in software program development projects. "

Per claim 3:

-the shelving component is executed on at least one of a local development system or a centralized server or servers.

Parrish: See Column 7." 57-59, "A Project History is a database which maintains various drafts, or versions, of the Project and is located in one or more server nodes. "

Per claim 4:

-the version control component includes committed work or files that are generally checked in as

Parrish: See Column 16 ....the History Server table entry of each History Server which stores any newly created or changed components is set to a 'NeedToCommit' state... If all member transactions reach the commit state, then the state entry for that History Server in the History Server Table is set to 'Committed. '"

Per claim 5:

-the version control system enables authorized users to retrieve respective versions of code to produce intermediate versions or builds of software in cases that utilize the shelving component or final versions or builds of software in cases that build from the committed work.

Parrish: See Column 17: 13-16, "After connection to the History Server which maintains the history of a given project, the Project workspace is empty and it is necessary to retrieve previous drafts of the program components to begin using the workspace. "

Per claim 6:

-the shelving component is employed for at least one of an interrupted workflow application, a checkpoint application, a shared work application, a code backup application, a work exchange application, and a private workspace exchange application.

Parrish: See Column 3: 21-24, "Accordingly, it is an object of the present invention to provide a program development management system which supports the reliable sharing and reuse of objects and other program components by a program development team."

Per claim 7:

-an unshelving component that restores the private workspace to a state that was previously archived.

Sarma disclosed restoring state at p. 5, Table 2, "ChangesReverted".

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Parrish, using the teachings of Sarma, because Parrish (col. 3: 25-32) recognized the need for a program development management system which maintains configuration and revision information and which can store different code versions developed over time, using a distributed network. Likewise, Sarma (Abstract) disclosed a novel workspace awareness tool, providing developers with insight into other workspaces. One would be motivated to reveal relevant information in a distributed configuration management to improve understanding among developers.

Per claim 8:

Parrish failed to explicitly disclose:

-the unshelving component allows removing the state that is stored on a server or preserving changes shelved on the server in order that the changes are available for unshelving by other users.

However, Sarma discloses (p. 1, lower left col. – upper right col.) a lock that prevents other developers from making concurrent modifications. (P. 2, right col. 3rd paragraph), locking is used to coordinate activities. (P. 3, left col, 2nd paragraph), available information is restricted to

which artifacts may potentially change (because they are locked)... See p. 5, Table 2, regarding events related to states.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Parrish, using the teachings of Sarma, because Parrish (col. 3: 25-32) recognized the need for a program development management system which maintains configuration and revision information and which can store different code versions developed over time, using a distributed network. Likewise, Sarma (Abstract) disclosed a novel workspace awareness tool, providing developers with insight into other workspaces. One would be motivated to reveal relevant information in a distributed configuration management to improve understanding among developers. Locking an artifact is a known technique to prevent unauthorized changes.

Per claim 9:

-a component to store state information for all files or folders in [[a]] the private workspace or for individual files in the private workspace.

Parrish: Col. 7: 57-59 Project History database maintains drafts, versions

Per claim 10:

-the state information for all files and folders includes at least one of a unique identifier for all files and folders in the workspace and a version number of each file or folder in the workspace.

Parrish: Col. 9: 27-29, unique ids

Sarma disclosed 'artifacts' (files and folders) retrieved from a repository (p. 1, right column).

Per claim 11:

-the state information for individual files includes at least one of a unique identifier for an individual file, a number of a version that was modified to create a current state, and a locally modified state of a file.

Parrish: Col. 9: 27-29, unique Ids

Per claim 12:

-storing meta-data that is associated with the state information.

Parrish: Col. 9: 59-60, metadata

Per claim 13:

Parrish failed to explicitly disclose:

-the unshelving component merges unshelved changes with changes pending on the private workspace when an unshelve operation is initiated.

However, Sarma discloses 'merging tools' at p. 1, right col., 1st paragraph. P. 2, right col., 4th paragraph, "coordinate parallel activities via the use of merge tools that combine changes to an artifact by one developer with changes to the same artifact by another developer..."

It would have been obvious, to one of ordinary skill in the art, at the time of the invention to include 'merging tools' because they are known (p. 2, 4th paragraph) in 'optimistic configuration management systems'. The coordinate parallel activities and resolve conflicts.

6. Claims 14, 15, & 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over USPN 6,385,765 to Ziebell, in view of "Palantir: Raising Awareness among Configuration Management Workspaces", by Anita Sarma, Zahra Noroozi, and Andre van der hock (2003 IEEE) (hereinafter Sarma).

Per claim 14:

A code development system embodied on a computer-readable storage medium, comprising:

-means for developing pending changes on non-finalized software on a private workspace;

-means for archiving the non-finalized software with pending changes m to a version control system;

-means for capturing one or more states associated with the non-finalized software;

-means for removing the pending changes from the non-finalized software on the private workspace;

-means for processing the archived non-finalized software along with finalized software on the version control system.

Ziebell (USPN 6385768) discloses

Column 6:32-35, "... a check-in function in VCS allows a developer to create a new revision in the archive from the revision that was previously checked out by the developer. " Column 5: 34-35, "A version identifier in VCS identifies each revision in VCS." Column 5: 53-54, "The revisions included can belong to one or more archives. "

More explicitly Sarna discloses:

-means for developing pending changes...means for archiving the non-finalized software with pending changes and means for capturing one or more states...means for removing the pending changes. See p. 4, right col., Sarma discloses events and related states that are captured in the workspace activities. Such events include: Populated, ChangesInProgress, ChangesCommitted, and UnPopulated, Synchroinzed, Added, Removed, Renamed, Moved. Also see Table 2, page 5.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Ziebell, using the teachings of Sarma, because one would be motivated to coordinate all events in a distributed development configuration system.

Per claim 15:

Ziebell failed to explicitly disclose:

-means for restoring the states and the non-finalized software to a previous state.

However, Sarma disclosed 'ChangesReverted' in Table 2, page 5.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Ziebell, using the teachings of Sarma, because one would be motivated to coordinate all events in a distributed development configuration system.

Per claim 22:

A user interface embodied on a computer-readable storage medium for archiving intermediate versions of code, comprising:

-a display component to highlight one or more files on a private workspace to archive on a version control system, the files contain pending changes associated with an intermediate version of code;

-a shelving command input to enable users to archive the intermediate files with pending changes to a version control system as if the intermediate files were finalized versions of the intermediate files,

-archiving removes the pending changes from the intermediate files on the private workspace.

Ziebell discloses:

See FIG. 1 (user interface embodied on a computer-readable storage medium for archiving intermediate versions of code) Column 5: 50-62, "A circle 27 represents Archive 1, a circle 28 represents a change that includes one or more revisions shown by blocks 29, 30 and 31," and, a circle 32 represents Archive II. The revisions included in a change can belong to one or more archives. Thus, revisions 29 and 30 are from Archive I while revision 31 is from Archive 11. The change keeps track of the Archives and the revisions from those Archives that were modified as part of the change. A change can be associated with a workspace. Each time a revision is checked out to the workspace it is added along with its archive to the change. In this way, VCS knows what Archives and what revisions from the Archive that make up a change." Column 5: 58-62, "Each time a revision is checked out to the workspace it is added along with its archive to the change. In this way, VCS knows what Archives and what revisions from the Archive that make up a change. " Column 7." 5-6, "After being tested, inspected and approved, a change will be ready to be incorporated into a software release. (intermediate files were finalized versions of the intermediate files)."

Ziebell failed to explicitly disclose:

-events and related states in a workspace.

However Sarma discloses: Table 2, page 5 discloses such events as Synchronized (finalized version), ChangesInProgress, ChangesCommitted, etc.

Note that a developer may archive a changed artifact, and return to further modify at a later time (archive intermediate files).

Therefore, it would have been obvious, to one of ordinary skill in the art at the time of the invention to modify Ziebell, to include events and states related to configuration because the additional information provided to developers working in a parallel distributed manner helps provide consistency and avoid conflicts.

7. Claims 16-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over USPN 5,752,245 to Parrish, in view of "Palantir: Raising Awareness among Configuration Management Workspaces", by Anita Sarma, Zahra Noroozi, and Andre van der hock (2003 IEEE) (hereinafter Sarma), and further in view of USPN 7,272,625 B1 to Hannel et al.

Per claim 16:

A method to facilitate code development, comprising:

-creating a version of software in a private development system;

-automatically determining at least one state for the software;

-shelving the software and the state on a version control system;

-unshelving the version of software to a private workspace in accordance with the state, the version of software is deleted from the version control system when unshelving is initiated by an owner of the software, and the version of software is preserved on the version control system when unshelving is initiated by a non-owner of the software.

Parrish discloses (col. 7: 57-59) a Project History database, maintains drafts & versions.

More explicitly Sarma disclosed events and states of the configuration system.

However, Sarma disclosed a "configuration management workspace awareness tool" (p. 1, right col., 3rd paragraph). Sarma disclosed workspace events (page 4, right col.) that describe ongoing activities in each workspace. Such states disclosed are POPULATED, CHANGESIN PROGRESS, CHANGESCOMMITTED, and UNPOPULATED. "The pair ChangesInProgress and ChangesCommitted may be repeated if a developer continues to make changes to the artifact before removing it from the workspace. (remove pending changes from software design on the private workspace after the current state has been captured). Sarma disclosed other events that may be captured (p. 4, right col, last paragraph), SYNCHRONIZED, ADDED, REMOVED, RENAMED, MOVED.

Parrish / Sarma failed to disclose permissions / access controls by 'owners' or 'non-owners'.

However Hannel further provided disclosure related to (col. 5: 64 – col. 6: 1) "policy enforcement system...policy server that includes an extensible policy database and a policy enforcer." Col. 6: 35-43, "conditions may be attached to policies in the policy database of the policy enforcement system...temporal conditions...a policy may be made which defines access to information and a temporal condition on the policy may restrict the policy's validity...

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Parrish, using the teachings of Sarma, because Parrish (col. 3: 25-32) recognized the need for a program development management system which maintains configuration and revision information and which can store different code versions developed over time, using a distributed network. Likewise, Sarma (Abstract) disclosed a novel workspace awareness tool, providing developers with insight into other workspaces. One would be motivated to reveal relevant information in a distributed configuration management to improve understanding among developers. Sarma discloses (p. 5, right col.), "must distinguish an artifact that is in a workspace in its original, repository-equivalent state from that same artifact in that same workspace after it has undergone some changes" "The name and version of an artifact are an integral part of the Palantir artifact identifier. To distinguish an artifact that has changed from one that has not, the qualifiers WS and REP are used...Finally, to distinguish different workspaces, a unique author identifier is used (owner / non-owner identified)..." It would have been obvious to further modify Parrish / Sarma to include database access policies, as disclosed by Hannel. One would be motivated to enforce restrictive actions and use to control security and consistency of information.

Per claim 17:

Parrish failed to explicitly disclose:

-unshelving the version of software to a private workspace that contains other pending changes causes the pending changes to be merged with the unshelved version of software.

However, Sarma discloses 'merging tools' at p. 1, right col., 1st paragraph. P. 2, right col., 4th paragraph, "coordinate parallel activities via the use of merge tools that combine changes to an artifact by one developer with changes to the same artifact by another developer..."

It would have been obvious, to one of ordinary skill in the art, at the time of the invention to include 'merging tools' because they are known (p. 2, 4th paragraph) in 'optimistic configuration management systems'. The coordinate parallel activities and resolve conflicts.

Per claim 18:

-providing at least one of a shelving command and an unshelving command to facilitate development of the software.

Parrish: Col. 7: 1-3, user interface with commands for a configuration workspace.

Per claim 19:

-the shelving command is associated with at least one of a preserve option, a workspace option, a replace option, a comment option, a file option, an error condition, and an exit code.

Parrish: Col. 7:1-3, user interface with commands for a configuration workspace

Per claim 20:

Parrish / Sarma failed to explicitly disclose:

-the shelving and the unshelving command are associated with at least one security parameter.

However, Hannel discloses database policies (col. 6: 30-43) related to permissions and conditions. Hannel discloses encryption and authentication at col. 8: 50-58.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Parrish, using the teachings of Sarma, because Parrish (col. 3: 25-32) recognized the need for a program development management system which maintains configuration and revision information and which can store different code versions developed over time, using a distributed network. Likewise, Sarma (Abstract) disclosed a novel workspace awareness tool, providing developers with insight into other workspaces. One would be motivated to reveal relevant information in a distributed configuration management to improve understanding among developers. Sarma discloses (p. 5, right col.), "must distinguish an artifact that is in a workspace in its original, repository-equivalent state from that same artifact in that same workspace after it has undergone some changes" "The name and version of an artifact are an integral part of the Palantir artifact identifier. To distinguish an artifact that has changed from one that has not, the qualifiers WS and REP are used...Finally, to distinguish different workspaces, a unique author identifier is used (owner / non-owner identified)..." It would have been obvious to further modify Parrish / Sarma to include database access policies, as disclosed

by Hannel. One would be motivated to enforce restrictive actions and use to control security and consistency of information.

Per claim 21:

-the unshelving command is associated with at least one of a preserve option, a file option, a name option, a username option, an error condition, and an exit code.

See Sarma, Table 2, p. 5, Rename (name option). See p. 8 5.2 CVS 'check out: Populated, edit: ChangesInProgress, update: Synchronized, commit: ChangesInProgress + SeverityChanged + ChangesCommitted (preserve option)... additions and removals are local to a workspace until the parent artifact is checked in.

8. Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over USPN 6385768 B1 to Ziebell, in view of "Palantir: Raising Awareness among Configuration Management Workspaces", by Anita Sarma, Zahra Noroozi, and Andre van der hock (2003 IEEE) (hereinafter Sarma), and further in view of USPN 7,272,625 to Hannel et al.

Per claim 23:

-an unshelving command that restores local versions of selected files on the version control system to a previous state,

-the selected files are deleted from the version control system when the unshelving command is initiated by an owner of the files,

-and the selected files are maintained on the version control system when the unshelving command is initiated by a non-owner of the files.

Sarma discloses (Table 2, page 5) various events including ChangesReverted (restoring), Removed (deleted) and related states.

Sarma failed to disclose permissions / access controls by 'owners' or 'non-owners'.

However Hannel further provided disclosure related to (col. 5: 64 – col. 6: 1) "policy enforcement system...policy server that includes an extensible policy database and a policy enforcer." Col. 6: 35-43, "conditions may be attached to policies in the policy database of the policy enforcement system...temporal conditions...a policy may be made which defines access to information and a temporal condition on the policy may restrict the policy's validity...

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, Sarma, because Parrish (col. 3: 25-32) recognized the need for a program

development management system which maintains configuration and revision information and which can store different code versions developed over time, using a distributed network. Likewise, Sarma (Abstract) disclosed a novel workspace awareness tool, providing developers with insight into other workspaces. One would be motivated to reveal relevant information in a distributed configuration management to improve understanding among developers. Sarma discloses (p. 5, right col.), "must distinguish an artifact that is in a workspace in its original, repository-equivalent state from that same artifact in that same workspace after it has undergone some changes" "The name and version of an artifact are an integral part of the Palantir artifact identifier. To distinguish an artifact that has changed from one that has not, the qualifiers WS and REP are used...Finally, to distinguish different workspaces, a unique author identifier is used (owner / non-owner identified)..." It would have been obvious to further modify Parrish / Sarma to include database access policies, as disclosed by Hannel. One would be motivated to enforce restrictive actions and use to control security and consistency of information.

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei

Zhen can be reached at (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned: 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

01/31/2008

MARY STEELMAN
PRIMARY EXAMINER